Contents

Kubernetes Gateway API

2 minute read ✓ page test

This task describes how to configure Istio to expose a service outside of the service mesh cluster, using the Kubernetes Gateway API. These APIs are an actively developed evolution of the Kubernetes Service and Ingress APIs.

Setup

1. Install the Gateway API CRDs:

   ```
   $ kubectl kustomize "github.com/kubernetes-sigs/gateway-api
   /config/crd?ref=v0.3.0" | kubectl apply -f -
   ```

2. Install Istio:

   ```
   $ istioctl install
   ```

3. Follow the instructions in the Determining the ingress IP and ports sections of the Ingress Gateways task in order to retrieve the external IP address of your

ingress gateway.

Configuring a Gateway

See the Gateway API documentation for information about the APIs.

1. Deploy a test application:

   ```
   $ kubectl apply -f @samples/httpbin/httpbin.yaml@
   ```

2. Deploy the Gateway API configuration:

```
$ kubectl apply -f - <<EOF
apiVersion: networking.x-k8s.io/v1alpha1
kind: GatewayClass
metadata:
  name: istio
spec:
  controller: istio.io/gateway-controller
---
apiVersion: networking.x-k8s.io/v1alpha1
kind: Gateway
metadata:
  name: gateway
  namespace: istio-system
spec:
  gatewayClassName: istio
  listeners:
  - hostname: "*"
    port: 80
    protocol: HTTP
    routes:
      namespaces:
        from: All
      selector:
        matchLabels:
          selected: "yes"
      kind: HTTPRoute
---
apiVersion: networking.x-k8s.io/v1alpha1
kind: HTTPRoute
metadata:
  name: http
  namespace: default
  labels:
    selected: "yes"
spec:
  gateways:
    allow: All
  hostnames: ["httpbin.example.com"]
  rules:
  - matches:
    - path:
        type: Prefix
        value: /get
    filters:
    - type: RequestHeaderModifier
      requestHeaderModifier:
        add:
          my-added-header: added-value
    forwardTo:
    - serviceName: httpbin
      port: 8000
EOF
```

3. Access the *httpbin* service using *curl*:

```
$ curl -s -I -HHost:httpbin.example.com "http://$INGRESS_HO
ST:$INGRESS_PORT/get"
HTTP/1.1 200 OK
server: istio-envoy
...
```

Note the use of the `-H` flag to set the *Host* HTTP header to “httpbin.example.com”. This is needed because the `HTTPRoute` is configured to handle “httpbin.example.com”, but in your test environment you have no DNS binding for that host and are simply sending your request to the ingress IP.

4. Access any other URL that has not been explicitly exposed. You should see an HTTP 404 error:

```
$ curl -s -I -HHost:httpbin.example.com "http://$INGRESS_HO
ST:$INGRESS_PORT/headers"
HTTP/1.1 404 Not Found
...
```